Discrete Optimization 9 (2012) 1–16

Contents lists available at SciVerse ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

Computer-assisted proof of performance ratios for the Differencing Method ☆

W. Michiels [a,*], E. Aarts [a], J. Korst [a], J. van Leeuwen [b], F.C.R. Spieksma [c]

[a] Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

[b] Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands

[c] KULeuven, Naamsestraat 69, 3000 Leuven, Belgium

ARTICLE INFO

Article history:
Received 24 November 2009
Received in revised form 26 September 2011
Accepted 5 October 2011
Available online 28 October 2011

Keywords:
Differencing Method
Balanced number partitioning
Worst-case performance
Computer-assisted proof
Mixed integer linear programming
Exact LP solver

ABSTRACT

We consider the problem of partitioning a set of n numbers into m subsets of cardinality $k = \lceil n/m \rceil$ or $\lfloor n/m \rfloor$, such that the maximum subset sum is minimal. We show that the performance ratio of the Differencing Method of Karmarkar and Karp for solving this problem is at least $2 - \sum_{i=0}^{k-1} \frac{i!}{k!}$ for any fixed k. We also build a mixed integer linear programming model (MILP) whose solution yields the performance ratio for any given k. For $k \leq 7$ these MILP-instances can be solved with an exact MILP-solver. This results in a computer-assisted proof of the tightness of the aforementioned lower bound for $k \leq 7$. For $k > 7$ we prove that $2 - \frac{1}{k-1}$ is an upper bound on the performance ratio, thereby leaving a gap with the lower bound of $\Theta(k^{-3})$, which is less than 0.4%.

1. Introduction

A classical problem in computer science is the number partitioning problem: given a set of n numbers, partition it into m subsets such that the maximum subset sum is minimal. For this problem, the Largest Differencing Method (LDM) of Karmarkar and Karp [1] outperforms other known polynomial-time approximation algorithms, such as LPT [2] and Multifit [3], from an average-case perspective. This claim is supported by empirical results [4] and theoretical results on its asymptotic performance [5,6,1,7,8].

A natural constraint on the solutions of the number partitioning problem is to demand that the cardinality of the subsets is balanced. This means that each subset contains either $k = \lceil n/m \rceil$ or $\lfloor n/m \rfloor$ numbers. The resulting problem is called balanced number partitioning. Yakir [8] introduced the Balanced Largest Differencing Method (BLDM) for solving this problem. Both BLDM and LDM are implementations of the same principle, called the Differencing Method. For $m = 2$ Yakir proves that the asymptotic average-case performance of BLDM is as good as that of LDM. Babel et al. [9] present several approximation algorithms for the balanced number partitioning problem. All these algorithms are outperformed by BLDM with respect to their average-case performance as we indicate in Section 3.

In this paper we analyze the worst-case performance ratios of BLDM for the balanced number partitioning problem. More precisely, we explicitly calculate the exact ratios for $k \leq 7$ using a computer-assisted technique, and we prove very tight bounds on the ratios for $k > 7$.

☆ This paper is a revised and improved version of [35].

* Corresponding author. Tel.: +31 40 27 47828.
E-mail address: wil.michiels@philips.com (W. Michiels).

1572-5286/$ – see front matter
doi:10.1016/j.disopt.2011.10.001

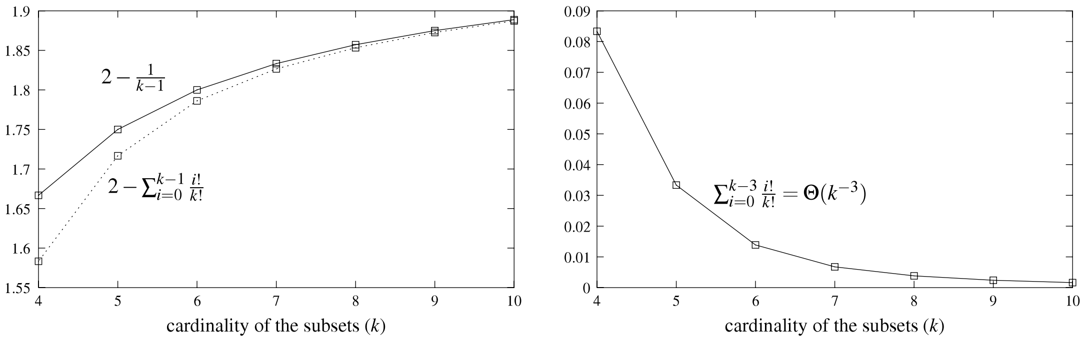

Fig. 1. On the left the lower and upper bounds on the performance ratio of BLDM as proved in this paper. On the right the difference between these bounds.

Balanced number partitioning. A problem instance I of the balanced number partitioning problem is specified by an integer m and a set $A = \{1, 2, \ldots, n\}$ of n items, where each item $j \in A$ has a nonnegative size a_j and $a_1 \le a_2 \le \cdots \le a_n$. Let $k = \lceil \frac{n}{m} \rceil$. The goal of the problem is to find a partition $\mathcal{A} = (A_1, A_2, \ldots, A_m)$ of A into m subsets of cardinality $k-1$ or k, such that $f_I(\mathcal{A})$ is minimal, where

$$f_I(\mathcal{A}) = \max_{1\le i\le m} S(A_i)$$

and $S(A_i) = \sum_{j\in A_i} a_j$.

As applications of the balanced number partitioning problem, Tsai [10] mentions the allocation of component types to pick-and-place machines for printed circuit board assembly [11] and the assignment of tools to machines in flexible manufacturing systems [12]. McDiarmid [13] finds an application of the problem in solving a cutting stock problem.

Both the number partitioning problem and the balanced number partitioning problem are strongly NP-hard [14]. Nevertheless, Hochbaum and Shmoys [15] show that the number partitioning problem admits a PTAS. For fixed k, the balanced number partitioning problem can be solved by a similar PTAS. However, no PTAS is known for the case that k is part of the input.

Because of the practical relevance of BLDM resulting from its good average-case performance, it is of interest to study its worst-case performance. Besides giving a performance guarantee, a worst-case analysis enhances our understanding of the algorithm. The analysis in this paper also gives a nice example of a computer-assisted proof. The most famous example of such a proof is the proof of the four color theorem by Appel et al. [16–18]. In this paper we formulate the problem of finding the performance ratios of BLDM for any given k as a mixed integer linear programming (MILP) problem. Based on this we can determine the performance ratio for any $k \le 7$ as for these values of k the size of the problem allows us to solve it with a computer by an exact MILP solver. For $k > 7$ the problem remains too large to be solved by an exact solver. Related to our approach is the paper of Zwick [19] who proves the performance ratio of a MAX 3-SAT approximation algorithm by a computer-assisted proof. However, his proof does not use an exact MILP or LP solver. Examples of computer-assisted proofs that make use of an LP solver (but not for proving a performance ratio) are given by Coffman et al. [20], Bubley et al. [21], and Hales [22]. Of these proofs, the one given by Hales is probably the most well-known. It proves the long-standing Kepler conjecture about sphere packing.

Outline of the paper. This paper is organized as follows. We start in Section 2 with a discussion of related work. In Section 3 we present the BLDM algorithm. In Sections 4–6 we next analyze the worst-case performance of the algorithm as a function of k. In [4] we prove that if m is given instead of k, then the performance ratio of BLDM is precisely $2 - 1/m$. To prevent case distinctions that distract from the essence of the derivation, we restrict ourselves to $k \ge 4$. Nevertheless, a similar analysis can be given for $k = 3$ resulting in a performance ratio of $4/3$ [4]. For $k = 2$, the problem is easy as can be seen as follows. Assume without loss of generality that we are given a problem instance with $2m$ items (otherwise, we construct such a problem instance by adding one item with size zero). Then one can prove that an optimal partition is obtained by assigning the $n/2$ smallest items increasingly and the $n/2$ largest items decreasingly to $A_1, A_2, \ldots, A_m$, i.e., $A_i = \{i, n-i+1\}$ for $1 \le i \le m$.

The worst-case analysis in Sections 4–6 is structured as follows. In Section 4 we prove that $2 - \sum_{i=0}^{k-1} \frac{i!}{k!}$ is a lower bound on the performance ratio of BLDM, and in Section 5 we show that $2 - \frac{1}{k-1}$ is an upper bound. The difference between these bounds decreases as k^{-3} for increasing k. The results are summarized in Fig. 1. The upper bound is proved by transforming the algorithm and showing that the set of relevant problem instances can be reduced. This analysis presented in Section 5 is extended in Section 6, where we formulate the problem of determining a performance ratio for any $k \ge 4$ as an MILP problem. By solving this MILP problem with the help of an exact MILP-solver, we prove that the lower bound of Section 4 is tight for $k \le 7$. We end with conclusions in Section 7.

2. Related work

Several approaches to solving the balanced number partitioning problem have been studied in the past. For $k = 3$ Kellerer and Woeginger [23] prove a worst-case performance ratio of $4/3 - 1/(3m)$ for the well-known Largest Processing Time (LPT) algorithm [2] adapted to the balanced number partitioning problem. An algorithm with a performance ratio of $7/6$ is presented for the case $k = 3$ by Kellerer and Kotov [24]. Babel et al. [9] analyze several approximation algorithms for the generic balanced number partitioning problem, i.e., for the case that k can be any value. A mixture of LPT and Multifit [3] achieves the best performance bound, namely 4/3. However, all algorithms presented by Babel et al. are outperformed by differencing methods such as BLDM and PDM with respect to their average-case performance.

The Differencing Method has been applied in several ways. Consider LDM when applied to number partitioning. Fischetti and Martello [25] prove that the algorithm has a performance ratio of 7/6 for $m = 2$. Furthermore, we prove in [26] that for $m \geq 3$ the performance ratio of LDM lies between $\frac{4}{3} - \frac{1}{3(m-1)}$ and $\frac{4}{3} - \frac{1}{3m}$. The two most popular polynomial-time algorithms for number partitioning are LPT and Multifit. It follows that although LDM has a better average-case performance than LPT and Multifit, Multifit outperforms LDM from a worst-case perspective. However, the worst-case performance of LDM is at least as good as that of LPT [2].

Yakir [8] proves that if $m = 2$ and the item sizes are uniformly distributed on $[0, 1]$, then the expected difference between the sum of the two subsets in a partition generated by either LDM or BLDM is $\mathcal{O}(n^{-c \log n})$ for some constant c. This implies that also the expected deviation of the cost of such a partition from the cost of an optimal partition, either balanced or not necessarily balanced, is $\mathcal{O}(n^{-c \log n})$. In [27,28,10], a probabilistic analysis is given for three alternative differencing methods for balanced number partitioning with $m = 2$. However, these algorithms have a worse average-case performance than BLDM.

For given $m \geq 2$, Karmarkar and Karp [1] present a rather elaborate differencing method that does, as LDM, not necessarily give a balanced partition. The algorithm uses some randomization in selecting the pair that is to be differenced so as to facilitate its probabilistic analysis. For the algorithm, they prove that the difference between the maximum and minimum sum of any subset is at most $\mathcal{O}(n^{-c \log n})$, almost surely, when the item sizes are in $[0, 1]$ and the density function is reasonably smooth. Tsai [7] proposes a modification of the algorithm that preserves this probabilistic result but enforces that balanced partitions are obtained.

Korf [29] presents a branch-and-bound algorithm, which starts with LDM and then tries to find a better solution until it ultimately finds an optimal solution to the number partitioning problem. By running BLDM instead of LDM and by modifying the search for better solutions, Mertens [30] changes the algorithm into an exact algorithm for balanced number partitioning. Although both algorithms are practically useful for $m = 2$, they have a time complexity that make them less interesting for $m > 2$.

3. Balanced Largest Differencing Method: the algorithm

In this section we discuss BLDM for any $m \geq 2$. If m and k do not divide n, then we add $l = m - (n \bmod m)$ items with size zero to get a problem instance with $n = mk$. As BLDM assigns these l items to different subsets, removing these items from the solution $\mathcal{A}^{\text{BLDM}}$ constructed by BLDM yields a solution for the original problem instance with the same cost as $\mathcal{A}^{\text{BLDM}}$. Hence, in our discussion of BLDM we assume that $n = mk$.

Let G_1 be the set containing the m smallest items, G_2 the set containing the m smallest remaining items, and so on. Hence, $G_i = \{(i-1)m + r \mid 1 \leq r \leq m\}$ for $1 \leq i \leq k$. Initially, BLDM starts with a sequence L of k partial solutions $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$, where $\mathcal{A}_i = \{A_{i1}, A_{i2}, \ldots, A_{im}\}$ is obtained by assigning each item of G_i to a different subset. More precisely, $A_{ij} = \{(i-1)m + j\}$. Next, the algorithm selects two partial solutions from L for which $d(\mathcal{A})$ is largest, where $d(\mathcal{A})$ is defined as the difference between the maximum and minimum subset sum in $\mathcal{A}$. These two solutions, denoted by $\mathcal{A}'$ and $\mathcal{A}''$, are combined into a new partial solution $\mathcal{A}$ by joining the subset with the smallest sum in $\mathcal{A}'$ with the subset with the largest sum in $\mathcal{A}''$, the subset with the second smallest sum in $\mathcal{A}'$ with the subset with the second largest sum in $\mathcal{A}''$, and so on. This process is called *differencing* $\mathcal{A}'$ and $\mathcal{A}''$. Hence, $\mathcal{A}$ is formed by the m subsets $A'_j \cup A''_{m-j+1}$ with $1 \leq j \leq m$, where the subsets of $\mathcal{A}'$ and $\mathcal{A}''$ have been ordered by non-decreasing sum. This solution replaces $\mathcal{A}'$ and $\mathcal{A}''$ in L. We iterate this differencing operation until only one solution in L remains, which is the balanced solution $\mathcal{A}^{\text{BLDM}}$ returned by BLDM. Note that if m and k do not divide n in the original problem instance, then $\mathcal{A}^{\text{BLDM}}$ indeed has the claimed property that the l items with size zero that we introduced are assigned to different subsets.

In this paper, we define sets of items by giving the sequence of the sizes instead of the item indices. For example, instead of $A = \{1, 2, 3\}$ with $a_i = 5 + i$ we write $A = 6, 7, 8$. Furthermore, $A_1 - A_2 - \cdots - A_m$ denotes the partition $\{A_1, A_2, \ldots, A_m\}$ and A_j^l is a short-hand notation for the partition $A_j - A_j - \cdots - A_j$ (l times). We illustrate the algorithm by means of an example.

Let $m = 3$, $k = 4$, and the twelve item sizes be $1, 2, 4, 5, 8, 10, 11, 11, 16, 17, 17, 21$. Initially L consists of the partial solutions $\mathcal{A}_1 = 1 - 2 - 4$, $\mathcal{A}_2 = 5 - 8 - 10$, $\mathcal{A}_3 = 11 - 11 - 16$, and $\mathcal{A}_4 = 17 - 17 - 21$. As $d(\mathcal{A}_1) = 3$, $d(\mathcal{A}_2) = 5$, $d(\mathcal{A}_3) = 5$, and $d(\mathcal{A}_4) = 4$, the first iteration of BLDM replaces $\mathcal{A}_2$ and $\mathcal{A}_3$ in L by partial solution $\mathcal{A}_5 = (5, 16) - (8, 11) - (10, 11)$ with $d(\mathcal{A}_5) = 2$. In the next two iterations $\mathcal{A}_1$ and $\mathcal{A}_4$ are replaced by $\mathcal{A}_6 = (1, 21) - (2, 17) - (4, 17)$ and $\mathcal{A}_5$ and $\mathcal{A}_6$ by $\mathcal{A}^{\text{BLDM}} = (2, 5, 16, 17) - (1, 8, 11, 21) - (4, 10, 11, 17)$, which is the final balanced solution. This solution is not optimal as in the solution the three subsets have a sum of 40, 41, and 42, respectively, whereas in solution $(1, 8, 11, 21) - (2, 5, 17, 17) - (4, 10, 11, 16)$ the sums of the three subsets are all 41.

In the remainder of this paper we analyze the worst-case performance of BLDM. In our analysis we say that an algorithm has a *performance bound* U, if it always delivers a solution with a cost at most U times the optimal cost. If bound U is tight, then U is called a *performance ratio*. In the analysis the following lemma will be useful. The lemma says roughly that in each iteration BLDM constructs a solution that is at least as good as the worst of the two solutions from which it is constructed. For the proof of the result we refer to Karmarkar and Karp [1].

Lemma 1. *Let partition $\mathcal{A}$ be obtained by differencing partitions $\mathcal{A}'$ and $\mathcal{A}''$. Then $d(\mathcal{A}) \leq \max(d(\mathcal{A}'), d(\mathcal{A}''))$.*

The main value of BLDM is in its superior average-case performance compared to other approximation algorithms. To substantiate its superior average-case performance, we performed the following empirical analysis. We let the numbers k and m range from 1 to 25 and from 1 to 200, respectively. For each value of the pair (k, m) we randomly generated 1,000 problem instances by drawing $n = k \cdot m$ item sizes uniformly at random from the interval $[0, 1)$. To each problem instance, we applied BLDM and the algorithms proposed by Babel et al. [9]. Babel et al. present three approximation algorithms, called the folding algorithm, MLPT, and PD. The last algorithm is their champion algorithm from a worst-case perspective. It has a worst-case performance ratio of 4/3. The algorithm finds a solution by packing the items into bins of capacity 4/3 times a lower bound. Consequently, its average-case performance ratio is close to its worst-case performance ratio of 4/3 and therefore shows the worst average-case performance. For each pair of values (k, m), BLDM performed substantially better than the other algorithms. For instance, for $k = 10$ and $k = 25$ the deviation from the obvious lower bound $S(A)/m$ is at least six and at least 20 times smaller for BLDM than for the other algorithms, respectively. Besides approximation algorithms, Babel et al. also present a local search algorithm based on a swap neighborhood function. In our simulations, this algorithm, which does not necessarily run in polynomial time, outperformed BLDM for odd values of k and m large enough. For even values of k, BLDM showed a better performance.

4. A lower bound on the performance ratio

We start our analysis by showing that a lower bound for the performance ratio of BLDM is given by the formula $lb(k) = 2 - \sum_{i=0}^{k-1} \frac{i!}{k!}$ for any $k \geq 4$. We prove this lower bound by presenting a family $\{I_k \mid k \geq 4\}$ of problem instances with a performance ratio of at least $lb(k)$. In Sections 5 and 6 we show that this lower bound equals the performance ratio for $k \leq 7$, while for $k > 7$ we will derive an upper bound that is very close to this lower bound.

Let $x_1, x_2, \ldots, x_k$ be k values with $x_{i-1} \leq x_i$ for $i = 2, \ldots, k$. Furthermore, consider a problem instance with $m \geq 2$, in which x_k occurs once, x_{k-1} occurs $2m - 1$ times, and x_i occurs m times for $i = 1, \ldots, k - 2$. Then it can be verified that BLDM returns the solution $\mathcal{A}^{\text{BLDM}} = (x_k, x_{k-1}, \ldots, x_1) - (x_{k-1}, x_{k-1}, x_{k-2}, \ldots, x_1)^{m-1}$. The cost of this solution is given by the sum of the first subset, i.e., by $\sum_{i=1}^{k} x_i$. This holds regardless of the values of x_i and m. We now assign values to x_i and m that depend on k, to obtain the problem instance I_k.

$$x_i^{(k)} = \begin{cases} 0 & \text{if } i = 1, \\ \dfrac{k!}{2} & \text{if } i = k, \\ \dfrac{\sum_{j=1}^{i-1} (k-j)!}{2} & \text{otherwise.} \end{cases} \tag{1}$$

and

$$m^{(k)} = 1 + \sum_{i=2}^{k-1} x_i = 1 + \frac{1}{2} \sum_{i=2}^{k-1} \sum_{j=1}^{i-1} (k-j)!. \tag{2}$$

To see that this problem instance indeed has a performance ratio of at least $lb(k)$, we present a solution $\mathcal{A}$ with $f_{I_k}(\mathcal{A}^{\text{BLDM}})/f_{I_k}(\mathcal{A}) = lb(k)$. For ease of notation, we will ignore the superscript "(k)" in $x_i^{(k)}$ and simply write x_i. For $i = 1, \ldots, k - 2$ we define subset V_i by

$$V_i = (x_1, x_2, \ldots, x_{k-i-2}, x_{k-i}, x_{k-i}, \ldots, x_{k-i}). \tag{3}$$

Hence, V_i contains $i + 2$ elements of size x_{k-i} and is next filled up to k elements by adding $x_1, x_2, \ldots$. Furthermore, let $z_i^{(k)}$ with $1 \leq i \leq k - 2$ be defined by

$$z_i^{(k)} = \begin{cases} \dfrac{2m^{(k)} - 1}{3} & \text{if } i = 1, \\ \dfrac{k! - \sum_{j=i}^{k-1} j!}{(i+2)i!} & \text{otherwise.} \end{cases} \tag{4}$$

We define solution $\mathcal{A}$ as

$$\mathcal{A} = (x_k, x_1, x_1, \ldots, x_1) - V_1^{z_1^{(k)}} - V_2^{z_2^{(k)}} - \cdots - V_{k-2}^{z_{k-2}^{(k)}}. \quad (5)$$

The following lemma, which is proved in Appendix, states that $\mathcal{A}$ defines a feasible solution for problem instance I_k. This means that the values $z_i^{(k)}$ sum up to $m^{(k)} - 1$, that they are all integrals, and that each variable x_i has the correct number of occurrences.

Lemma 2. *Solution $\mathcal{A}$ defined by (5) is a feasible solution.*

We now prove $f_{I_k}(\mathcal{A}^{\text{BLDM}})/f_{I_k}(\mathcal{A}) = lb(k)$. As a first step toward this result we prove the following lemma.

Lemma 3. *Let V_i be given by (3). Then we have $S(V_i) = x_k$ for each $1 \le i \le k-2$.*

Proof. From the definition of V_i it follows that $S(V_i)$ is given by

$$\left(\sum_{j=1}^{k-i-2} x_j\right) + (i+2)x_{k-i}.$$

Using the definition of x_i, the two terms in this equation can be rewritten to

$$\sum_{j=1}^{k-i-2} x_j = \frac{1}{2}\sum_{j=1}^{k-i-3}(k-j-(i+2))(k-j)!$$

and

$$(i+2)x_{k-i} = \frac{1}{2}\sum_{j=1}^{k-i-1}(i+2)(k-j)!$$

respectively. Combining these results yields that

$$2 \cdot S(V_i) = \left(\sum_{j=1}^{k-i-3}(k-j)(k-j)!\right) + (i+2)(i+2)! + (i+2)(i+1)!. \quad (6)$$

Using that $a \cdot a! = (a+1)! - a!$ for any a, it follows that $(i+2)(i+2)! = (i+3)! - (i+2)!$ and that for any ℓ we have

$$\sum_{j=1}^{\ell}(k-j)(k-j)! = k! - (k-\ell)!. \quad (7)$$

Hence, (6) can be rewritten as

$$S(V_i) = \frac{k!}{2}.$$

Since $x_k = \frac{k!}{2}$ this proves the lemma. □

It now follows that

$$\frac{f_{I_k}(\mathcal{A}^{\text{BLDM}})}{f_{I_k}(\mathcal{A})} = \frac{\sum_{i=1}^{k} x_i}{x_k}.$$

The following result states that this ratio is given by $lb(k)$.

Lemma 4. *Let x_i be given by (1). Then we have*

$$\frac{\sum_{i=1}^{k} x_i}{x_k} = 2 - \sum_{i=0}^{k-1}\frac{i!}{k!}. \quad (8)$$

Proof. Using the definition of x_i we get

$$\frac{\sum_{i=1}^{k} x_i}{x_k} = 1 + \frac{\sum_{i=2}^{k-1} x_i}{x_k} = 1 + \frac{\sum_{i=1}^{k-2}(k-i-1)(k-i)!}{k!}. \quad (9)$$

We now focus on the sum $\sum_{i=1}^{k-2}(k-i-1)(k-i)!$ in the numerator of the last expression. Using (7) we obtain

$$\begin{aligned}\sum_{i=1}^{k-2}(k-i-1)(k-i)! &= \left(\sum_{i=1}^{k-2}(k-i)(k-i)!\right) - \sum_{i=1}^{k-2}(k-i)! \\ &= k! - 2! - \sum_{i=1}^{k-2}(k-i)! \\ &= k! - \sum_{i=0}^{k-1} i!.\end{aligned}$$

Combining this result with (9) yields (8). □

Hence, we have proven the following result.

Theorem 5. *For given $k \geq 4$, a lower bound on the performance ratio of* BLDM *is given by*

$$2 - \sum_{i=0}^{k-1} \frac{i!}{k!}.$$

To illustrate the above, consider the case $k = 7$. Instance I_7 is then defined by

$$(x_7, x_6, x_5, x_4, x_3, x_2, x_1) = (2520, 436, 435, 432, 420, 360, 0)$$

and $m^{(7)} = 2084$. This implies that the solution $\mathcal{A}^{\text{BLDM}}$ returned by BLDM for this instance is given by

$$(2520, 436, 435, 432, 420, 360, 0) - (436, 436, 435, 432, 420, 360, 0)^{2083}.$$

The values $z_i^{(7)}$ are given by

$$(z_1^{(7)}, z_2^{(7)}, z_3^{(7)}, z_4^{(7)}, z_5^{(7)}) = (1389, 521, 139, 29, 5),$$

which results in the solution $\mathcal{A}$ defined by

$$\begin{aligned}&(2520, 0, 0, 0, 0, 0, 0) - (436, 436, 436, 432, 420, 360, 0)^{1389} \\ &\quad - (435, 435, 435, 435, 420, 360, 0)^{521} - (432, 432, 432, 432, 432, 360, 0)^{139} \\ &\quad - (420, 420, 420, 420, 420, 420, 0)^{29} - (360, 360, 360, 360, 360, 360, 360)^{5}.\end{aligned}$$

The cost of $\mathcal{A}^{\text{BLDM}}$ is given by the sum of the first subset, which is 4603. In $\mathcal{A}$ each subset has a sum of 2520 and thus also the cost of $\mathcal{A}$ is 2520. Hence,

$$f_{I_7}(\mathcal{A}^{\text{BLDM}})/f_{I_7}(\mathcal{A}) = 4603/2520 = lb(7).$$

5. An upper bound on the performance ratio

In this section, we prove that the performance ratio R is bounded from above by $2 - 1/(k-1)$ for $k \geq 4$. Since the value of k will be clear from the context, k is not explicitly given as index of ratio R, or instance I. Apart from proving an upper bound on R for any $k \geq 4$, this section also serves as a basis for Section 6.

Consider a problem instance I in which m and k do not divide n and let I' be obtained from I by adding $l = m - (n \bmod m)$ items with size zero. As indicated in Section 3, BLDM constructs a solution for I by constructing a solution for I' and by removing from this solution the l introduced items. Because the solutions for I and I' both have the same cost and because the cost of an optimal partition for I' is at most the cost of an optimal partition for I, we have that the performance ratio of BLDM for I' is at least the performance ratio of BLDM for I. Hence, without loss of generality we can assume in our analysis of the worst-case performance of BLDM that $n = mk$.

Let $k \geq 4$. By definition, U is a performance bound of BLDM if and only if for each instance I, $f_I(\mathcal{A}^{\text{BLDM}}) \leq U \cdot f_I^*$, where f_I^* denotes the cost of an optimal partition. For proving $f_I(\mathcal{A}^{\text{BLDM}}) \leq U \cdot f_I^*$, we may substitute $f_I(\mathcal{A}^{\text{BLDM}})$ by a larger expression and f_I^* by a smaller expression. The first two steps of the analysis given in this section concentrate on the nontrivial task of finding such expressions, where we successively change the algorithm and the problem instance. These steps are summarized in Fig. 2. We use this in the third step to prove lower and upper bounds on R.

In our analysis, we can disregard problem instances with a performance ratio at most 3/2 as for each $k \geq 4$ an instance with a ratio of 19/12 exists. For $k = 4$ this is the case as BLDM can derive partition $(12, 4, 3, 0) - (4, 4, 3, 0)^7$ with objective value 19, whereas an optimal partition of these numbers is $(12, 0, 0, 0) - (4, 4, 4, 0)^5 - (3, 3, 3, 3)^2$, which has objective value 12. Furthermore, when $k = 4 + i$ for $i \geq 1$ an instance with performance ratio of 19/12 can be obtained from this

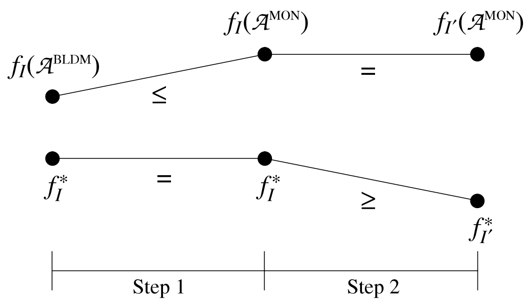

Fig. 2. Outline of the first two steps in Section 5.

instance by adding $i \cdot m$ items with size zero. Then in both the partition given by BLDM and an optimal partition, i items with size zero are added to each subset resulting in partitions with the same objective value.

Step 1: *construction of a partition* $\mathcal{A}^{\text{MON}}$ *with* $f_I(\mathcal{A}^{\text{BLDM}}) \leq f_I(\mathcal{A}^{\text{MON}})$. Partition $\mathcal{A}^{\text{BLDM}}$ has the property that each subset contains exactly one item from each set G_i. The same property holds for the so-called monotone partition $\mathcal{A}^{\text{MON}} = (A_1^{\text{MON}}, A_2^{\text{MON}}, \ldots, A_m^{\text{MON}})$. In this partition subset A_j^{MON} contains the jth largest item from G_k and the jth smallest items from $G_1, G_2, \ldots, G_{k-1}$, i.e., the subset contains item $n-j+1$ from G_k and item $(i-1)m+j$ from G_i with $1 \leq i < k$. For proving that indeed $f_I(\mathcal{A}^{\text{BLDM}}) \leq f_I(\mathcal{A}^{\text{MON}})$ holds, the following lemma will be useful. Note that we have $1 + \frac{m-1}{2m} < \frac{19}{12}$, which implies that Lemmas 6 and 7 cover all relevant problem instances.

Lemma 6. *If the performance ratio of* BLDM *for a given problem instance I is larger than* $1 + \frac{m-1}{2m}$, *i.e., if* $f_I(\mathcal{A}^{\text{BLDM}})/f_I^* > 1 + \frac{m-1}{2m}$, *then* $d(\mathcal{A}_k) > \sum_{i=1}^{k-1} d(\mathcal{A}_i)$.

Proof. We prove the lemma by contradiction. So, assume that $d(\mathcal{A}_k) \leq \sum_{i=1}^{k-1} d(\mathcal{A}_i)$ and $f_I(\mathcal{A}^{\text{BLDM}})/f_I^* > 1 + \frac{m-1}{2m}$. Remember that $\mathcal{A}_i$ with $1 \leq i \leq k$ is the initial solution of BLDM obtained by assigning each item of G_i to a different subset. We have

$$f_I^* \geq \frac{1}{m} \sum_{j=1}^{m} S(A_j^{\text{BLDM}}) \geq \min_{1 \leq j \leq m} S(A_j^{\text{BLDM}}) + \frac{d(\mathcal{A}^{\text{BLDM}})}{m}$$

and thus

$$f_I(\mathcal{A}^{\text{BLDM}}) = \min_{1 \leq j \leq m} S(A_j^{\text{BLDM}}) + d(\mathcal{A}^{\text{BLDM}}) \leq f_I^* + \frac{m-1}{m} d(\mathcal{A}^{\text{BLDM}}).$$

Hence, if we can prove $d(\mathcal{A}^{\text{BLDM}}) \leq f_I^*/2$, then $f_I(\mathcal{A}^{\text{BLDM}})/f_I^* \leq 1 + \frac{m-1}{2m}$ holds, which contradicts the assumption that $f_I(\mathcal{A}^{\text{BLDM}})/f_I^* > 1 + \frac{m-1}{2m}$. Thus, to prove the lemma it suffices to show that $d(\mathcal{A}^{\text{BLDM}}) \leq f_I^*/2$.

Let $\delta(G_i)$ be the difference between the smallest and largest item size in G_i. As item $n-m$ is the largest item in G_{k-1}, we obtain that a_{n-m} is at least $\sum_{i=1}^{k-1} \delta(G_i) = \sum_{i=1}^{k-1} d(\mathcal{A}_i)$. Since $d(\mathcal{A}_j) \leq \sum_{i=1}^{k-1} d(\mathcal{A}_i)$ for all $1 \leq j \leq k$, this implies $d(\mathcal{A}_j) \leq a_{n-m}$ for $1 \leq j \leq k$. Furthermore, we have $a_{n-m} \leq f_I^*/2$ because I contains $m+1$ items at least $n-m$, which implies that at least one subset in an optimal partition has to contain two items at least $n-m$. Combining these results yields $\max_{1 \leq i \leq k} d(\mathcal{A}_i) \leq f_I^*/2$. This implies $d(\mathcal{A}^{\text{BLDM}}) \leq f_I^*/2$ as $d(\mathcal{A}^{\text{BLDM}}) \leq \max_{1 \leq i \leq k} d(\mathcal{A}_i)$ by Lemma 1. □

Lemma 7. *If for a given problem instance I one has* $f_I(\mathcal{A}^{\text{BLDM}})/f_I^* > 1 + \frac{m-1}{2m}$, *then* $f_I(\mathcal{A}^{\text{BLDM}}) \leq f_I(\mathcal{A}^{\text{MON}})$.

Proof. First, we discuss a property of BLDM. Let permutation σ put the partial solutions $\mathcal{A}_i$ in order of non-increasing $d(\mathcal{A}_i)$-value. Hence, $d(\mathcal{A}_{\sigma(i)}) \geq d(\mathcal{A}_{\sigma(i+1)})$. Then, partial solutions $\mathcal{A}_{\sigma(1)}$ and $\mathcal{A}_{\sigma(2)}$ are differenced in the first iteration of BLDM, which results in $\mathcal{A}_{k+1}$. As $d(\mathcal{A}_k) > \sum_{i=1}^{k-1} d(\mathcal{A}_i)$ by Lemma 6, we have $\sigma(1) = k$. Moreover, since $d(\mathcal{A}_{k+1}) \geq d(\mathcal{A}_k) - d(\mathcal{A}_{\sigma(2)})$, the partial solution satisfies $d(\mathcal{A}_{k+1}) > \sum_{i=3}^{k} d(\mathcal{A}_{\sigma(i)})$. Based on this argumentation, one can prove by induction on i the more general statement that in the ith iteration of BLDM, partial solution $\mathcal{A}_{k+i-1}$ is differenced with $\mathcal{A}_{\sigma(i+1)}$. In other words, the final partition is obtained by successively adding the items of $G_{\sigma(2)}, G_{\sigma(3)}, \ldots, G_{\sigma(k)}$ to the partition $\mathcal{A}_k$ of G_k.

To prove the lemma, we now show by induction on i the following property for the partition $\mathcal{A}_{k+i} = \{A_1, A_2, \ldots, A_m\}$ obtained in iteration i. For each set A_j, a set A_l^{MON} in $\mathcal{A}^{\text{MON}}$ exists with $S(A_j) \leq S(A_l^{\text{MON}})$, where we only consider items in $\mathcal{A}^{\text{MON}}$ that are partitioned by $\mathcal{A}_{k+i}$. Hence, we only consider items from $G_{\sigma(1)}, G_{\sigma(2)}, \ldots, G_{\sigma(i+1)}$. To formalize the induction hypothesis, we first introduce some definitions.

For a set X of items, we define $X(i)$ as the subset containing only the items from $G_{\sigma(1)}, G_{\sigma(2)}, \ldots, G_{\sigma(i)}$. Correspondingly, $\mathcal{A}(i) = \{A_1(i), A_2(i), \ldots, A_m(i)\}$ for any solution $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. Note that $\mathcal{A}_{k+i} = \mathcal{A}^{\text{BLDM}}(i+1)$. We say that item j from G_i has *relative index* $r(j)$ if G_i contains exactly $r(j) - 1$ smaller items. Formally, this means $r(j) = j - (i-1)m$. For example, the relative index of an item j in A_l^{MON} is $m - l + 1$ if $j \in G_k$ and l, otherwise. The relative index $r(X)$ of a set X is

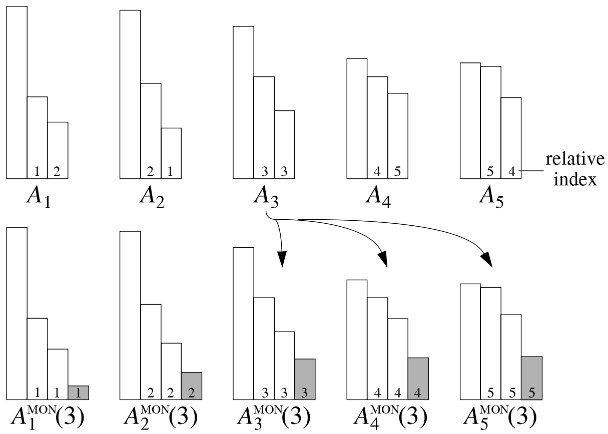

Fig. 3. Visualization of the induction hypothesis for $i = 3$ and $\sigma = (4, 3, 2, 1)$. The bars indicate the item sizes, and the shaded bars are not considered.

defined as the maximum relative index of any of its items, excluding the items from G_k. This implies that $r(A_l^{\text{MON}}) = l$. For $i \geq 1$, the induction hypothesis is now formalized as follows.

Induction hypothesis. For each A_j in $\mathcal{A}_{k+i} = \{A_1, A_2, \ldots, A_k\}$, an $l \geq r(A_j)$ exists, such that $S(A_l^{\text{MON}}(i+1)) \geq S(A_j)$.

The induction hypothesis is visualized for an example in Fig. 3. As the relative index of A_3 is 3, where A_3 is a subset in the partial solution $\mathcal{A}_6$ obtained in iteration $i = 2$ of BLDM, the induction hypothesis states that $S(A_3^{\text{MON}}(3))$, $S(A_4^{\text{MON}}(3))$, or $S(A_5^{\text{MON}}(3))$ is at least $S(A_3)$. Note that the induction hypothesis implies the interpretation given above.

Basis case. For $i = 1$, the induction hypothesis follows from the observation that $\mathcal{A}_{k+1} = \mathcal{A}^{\text{MON}}(2)$.

Induction step. Let $i \geq 2$, and let $A_j = \{\iota_1, \iota_2, \ldots, \iota_{i+1}\}$ be an arbitrary set from $\mathcal{A}_{k+i}$, where $\iota_p \in G_{\sigma(p)}$ for $1 \leq p \leq i+1$. To prove the induction hypothesis, we have to show that an $l \geq r(A_j)$ exists, such that $S(A_l^{\text{MON}}(i+1)) \geq S(A_j)$. Note that item ι_{i+1} has been added to A_j in iteration i of BLDM.

Assume that $r(\iota_{i+1})$ is not larger than the relative index of any other item in A_j not counting ι_1, i.e, $r(\iota_{i+1}) \leq r(A_j(i))$. By the induction hypothesis, a set $A_l^{\text{MON}}(i)$ exists with $l \geq r(A_j - \{\iota_{i+1}\})$, such that $S(A_l^{\text{MON}}(i)) \geq S(A_j - \{\iota_{i+1}\})$. As the item from $G_{\sigma(i+1)}$ in $A_l^{\text{MON}}(i+1)$ has relative index l and $l \geq r(\iota_{i+1})$, its size is at least the size of ι_{i+1}. Consequently, we obtain $S(A_l^{\text{MON}}(i+1)) \geq S(A_j)$ and $l \geq r(A_j)$, which proves the induction hypothesis for A_j with $r(\iota_{i+1}) \leq r(A_j(i))$.

Assume, on the other hand, that $r(\iota_{i+1}) > r(A_j(i))$. Then a set $A_{j'} = \{\iota'_1, \iota'_2, \ldots, \iota'_{i+1}\}$ with $\iota'_p \in G_{\sigma(p)}$ for $1 \leq p \leq i+1$ exists to which an item with at most the same relative index as ι_{i+1} has been assigned in iteration $i-1$ and to which an item with a smaller relative index has been assigned in iteration i, i.e., $r(\iota'_{i+1}) < r(\iota_{i+1}) \leq r(\iota'_i)$. This can be seen as follows. Assume that it is not the case. We show that this yields a contradiction. Then all $r(\iota_{i+1}) - 1$ subsets in $\mathcal{A}_{k+i}$ containing an item from $G_{\sigma(i+1)}$ with relative index strictly smaller than $r(\iota_{i+1})$ contain an item from $G_{\sigma(i)}$ with a relative index that is also strictly smaller than $r(\iota_{i+1})$. However, $G_{\sigma(i)}$ contains only $r(\iota_{i+1}) - 1$ such items of which one is assigned to A_j as $r(\iota_i) \leq r(A_j(i)) < r(\iota_{i+1})$. This yields a contradiction.

We return to proving the induction hypothesis for A_j. From the induction hypothesis, it follows that an $l \geq r(\iota'_i)$ exists, such that $S(A_l^{\text{MON}}(i)) \geq S(A_{j'}(i))$ holds. Furthermore, $r(\iota'_{i+1}) < r(\iota_{i+1})$ implies that, at the start of iteration i, the sum of $A_{j'}$ is at least the sum of A_j, i.e., $S(A_{j'}(i)) \geq S(A_j(i))$. Combining these two observations yields that $S(A_l^{\text{MON}}(i)) \geq S(A_j(i))$. As $l \geq r(\iota'_i) \geq r(\iota_{i+1})$, we have that the item from $G_{\sigma(i+1)}$ in $A_l^{\text{MON}}(i+1)$ is at least the item ι_{i+1} from $G_{\sigma(i+1)}$ in A_j. Hence, we get $S(A_l^{\text{MON}}(i+1)) \geq S(A_j)$ and $l \geq r(\iota_{i+1}) \geq r(A_j)$. This means that the induction hypothesis also holds for A_j in the case that $r(\iota_{i+1}) > r(A_j(i))$. This completes the proof of the lemma. □

Step 2: *construction of a new problem instance I' with $f_I(\mathcal{A}^{\text{MON}}) = f_{I'}(\mathcal{A}^{\text{MON}})$ and $f_I^* \geq f_{I'}^*$*. We construct instance I' from I by changing the item sizes. Let A_j^{MON} be the subset of $\mathcal{A}^{\text{MON}}$ with largest sum. Furthermore, let x_i with $1 \leq i \leq k$ be defined as the size of the only item that is both in A_j^{MON} and G_i. Note that $f_I(\mathcal{A}^{\text{MON}}) = \sum_{i=1}^{k} x_i$ and $x_1 \leq x_2 \leq \cdots \leq x_k$.

Instance I' is now constructed out of I by decreasing the size of each item i that is not in A_j^{MON} until it is either equal to the size of the largest item i' in A_j^{MON} with $i' < i$ or equal to zero; see Fig. 4. More precisely, subset A_l^{MON} with $l \neq j$ is set to $x_k, x_{k-2}, x_{k-3}, \ldots, x_1, 0$ if $l < j$ and to $x_{k-1}, x_{k-1}, x_{k-2}, \ldots, x_2, x_1$ otherwise. Hence, $\mathcal{A}^{\text{MON}}$ is given by

$$(x_k, x_{k-2}, x_{k-3}, \ldots, x_1, 0)^{j-1} - (x_k, x_{k-1}, x_{k-2}, \ldots, x_2, x_1) - (x_{k-1}, x_{k-1}, x_{k-2}, \ldots, x_2, x_1)^{m-j}.$$

As we only decrease sizes and do not affect the sizes in A_j^{MON}, both $f_I(\mathcal{A}^{\text{MON}}) = f_{I'}(\mathcal{A}^{\text{MON}})$ and $f_I^* \geq f_{I'}^*$.

Step 3: *bounding the performance ratio*. If we combine the results from Steps 1 and 2, we get $f_I(\mathcal{A}^{\text{BLDM}}) \leq f_{I'}(\mathcal{A}^{\text{MON}})$ and $f_I^* \geq f_{I'}^*$; see Fig. 2. Hence, if we prove for some U that $f_{I'}(\mathcal{A}^{\text{MON}}) \leq U \cdot f_{I'}^*$, then it follows that $f_I(\mathcal{A}^{\text{BLDM}}) \leq U \cdot f_I^*$, i.e., U is a performance bound for instance I. Consequently, to prove a performance bound U for BLDM, it suffices to prove that for

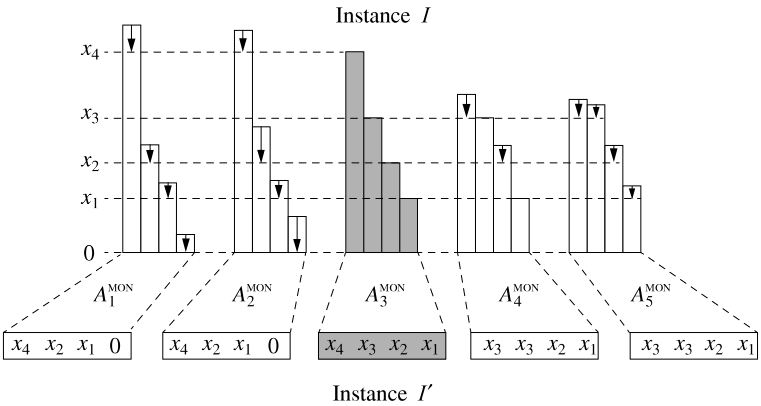

Fig. 4. Example of the construction of problem instance I' out of I, where A_j^{MON} is the subset in $\mathcal{A}^{\text{MON}}$ with largest sum.

each $x_k \geq x_{k-1} \geq \cdots \geq x_1 \geq 0$, $m_l \geq 0$, and $m_r \geq 0$, we have $f_I(\mathcal{A}^{\text{MON}}) \leq U \cdot f_I^*$, where I is defined such that $\mathcal{A}^{\text{MON}}$ is given by

$$(x_k, x_{k-2}, x_{k-3}, \ldots, x_1, 0)^{m_l} - (x_k, x_{k-1}, x_{k-2}, \ldots, x_2, x_1) - (x_{k-1}, x_{k-1}, x_{k-2}, \ldots, x_2, x_1)^{m_r}. \tag{10}$$

Using this observation, we can prove the theorem below. The theorem gives a performance bound for which it can be shown that it deviates from the lower bound of Theorem 5 by only

$$\sum_{i=0}^{k-3} \frac{i!}{k!} = \Theta(k^{-3}).$$

Theorem 8. *For given $k \geq 4$, an upper bound on the performance ratio of* BLDM *is given by* $2 - \frac{1}{k-1}$.

Proof. As indicated, it suffices to prove that

$$f_I(\mathcal{A}^{\text{MON}}) \leq \left(1 + \frac{k-2}{k-1}\right) f_I^*, \tag{11}$$

where I is defined such that $\mathcal{A}^{\text{MON}}$ is given by (10) for some $x_k \geq x_{k-1} \geq \cdots \geq x_1 \geq 0$, $m_l \geq 0$, and $m_r \geq 0$. Let ω_1, ω_2, and ω_3 be defined as the sum of the three types of subsets in $\mathcal{A}^{\text{MON}}$, i.e.,

$$\omega_1 = \omega_2 - x_{k-1}, \qquad \omega_2 = \sum_{i=1}^{k} x_i, \quad \text{and} \quad \omega_3 = \omega_2 - x_k + x_{k-1}.$$

Since all three ω_1, ω_2, and ω_3 cannot be strictly larger than f_I^* and since ω_2 is at least as large as both ω_1 and ω_3, we have $\omega_1 \leq f_I^*$ or $\omega_3 \leq f_I^*$. We consider these two cases separately.

Case 1: $\omega_1 \leq f_I^*$.
By definition, $f_I(\mathcal{A}^{\text{MON}}) = \omega_2 = \omega_1 + x_{k-1}$. Hence, (11) follows when $x_{k-1} \leq \frac{k-2}{k-1} f_I^*$. Since I contains $m_l + 2m_r + 2$ items with a size at least x_{k-1} and since these items have to be assigned to $m = m_l + m_r + 1$ subsets, an optimal partition has at least one subset containing two items with size at least x_{k-1}. Consequently, $x_{k-1} \leq \frac{1}{2} f_I^*$, which yields that $x_{k-1} \leq \frac{k-2}{k-1} f_I^*$ since $\frac{1}{2} \leq \frac{k-2}{k-1}$ for $k \geq 4$.

Case 2: $\omega_3 \leq f_I^*$.
If $\omega_3 = 0$, then $f_I(\mathcal{A}^{\text{MON}}) = x_k = f_I^*$, which clearly implies (11). Next, assume that $\omega_3 > 0$, which implies that $x_{k-1} > 0$. Since the number of items with size x_k is larger than the number of items with size 0, $x_k + x_1 \leq f_I^*$. Hence, for proving (11), it suffices to show that $x_{k-1} + x_{k-2} + \cdots + x_2 \leq \frac{k-2}{k-1} f_I^*$. By the definition of ω_3 and since $\omega_3 \leq f_I^*$, this is implied by

$$\frac{x_{k-1} + x_{k-2} + \cdots + x_2}{x_{k-1} + x_{k-1} + x_{k-2} + \cdots + x_1} \leq \frac{k-2}{k-1}.$$

It can be verified that the left-hand side is maximal whenever $x_1 = 0$ and $x_i = x_{k-1}$ for $2 \leq i < k$. Hence, the left-hand side is at most $\frac{(k-2)x_{k-1}}{(k-1)x_{k-1}} = \frac{k-2}{k-1}$. This proves the performance bound $2 - \frac{1}{k-1}$ for BLDM. □

6. Computer-assisted determination of the performance ratios

In Step 3 of Section 5 we completed the proof of the essential fact that

$$\sup_{(\mathcal{A},I)\in\Gamma^{(1)}} \frac{f_I(\mathcal{A}^{\mathrm{MON}})}{f_I(\mathcal{A})} \tag{12}$$

is a performance bound of BLDM for given $k \geq 4$. Here we have $(\mathcal{A}, I) \in \Gamma^{(1)}$ if and only if $\mathcal{A}$ is a solution of I and problem instance I has the property that values $0 \leq x_1 \leq x_2 \leq \cdots \leq x_k$, $m_l \geq 0$, and $m_r \geq 0$ exist such that partition $\mathcal{A}^{\mathrm{MON}}$ is given by (10). It can be proved that for determining (12) we only have to consider the instances of $\Gamma^{(2)} \subseteq \Gamma^{(1)}$ satisfying the following additional constraints.

- $m_l = 0$, i.e., $\mathcal{A}^{\mathrm{MON}} = (x_k, x_{k-1}, \ldots, x_1) - (x_{k-1}, x_{k-1}, x_{k-2}, \ldots, x_1)^{m-1}$.
- the subset in $\mathcal{A}$ containing x_k does, besides x_k, only contain zeros.
- $m \geq k-1$. Hence, we can assume that $A_1 = x_k, 0, 0, \ldots, 0 = x_k, x_1, x_1, \ldots, x_1$.

That is, the performance bound of BLDM given by (12) equals

$$\sup_{(\mathcal{A},I)\in\Gamma^{(2)}} \frac{f_I(\mathcal{A}^{\mathrm{MON}})}{f_I(\mathcal{A})}. \tag{13}$$

For the proof we refer to Michiels [4].

For an $(\mathcal{A}, I) \in \Gamma^{(2)}$, partition $\mathcal{A}^{\mathrm{MON}} = (x_k, x_{k-1}, \ldots, x_1) - (x_{k-1}, x_{k-1}, x_{k-2}, \ldots, x_1)^{m-1}$ is the same partition as the one given by BLDM. Furthermore, $f_I(\mathcal{A}) \geq f_I^*$. This implies that the performance ratio of BLDM for I is at least $\frac{f_I(\mathcal{A}^{\mathrm{MON}})}{f_I(\mathcal{A})}$. Hence, the performance bound given by (13) is tight, which means that, we did not lose tightness in our derivation.

We now formulate the problem of computing the performance ratio given by (13) as an MILP problem. By solving this problem computationally with the help of an exact MILP-solver, we then obtain performance ratios for $k = 4, 5, 6$ and 7. These performance ratios are equal to the lower bound derived in Section 4. Let us first introduce some definitions.

We define $\mathcal{W}$ as the set of all $(k-1)$-tuples $\bar{b}_j = (b_{1j}, b_{2j}, \ldots, b_{k-1,j}) \in \mathbb{N}^{k-1}$ with $\sum_{i=1}^{k-1} b_{ij} = k$. Let these tuples be numbered from 1 to t. Let $\bar{b}_j \in \mathcal{W}$ define the subset V_j containing b_{ij} items with size x_i for $1 \leq i < k$. For example, for $k = 5$ tuple $\bar{b}_j = (1, 2, 1, 1)$ defines subset $V_j = x_1, x_2, x_2, x_3, x_4$. Note that x_k does not occur in a subset corresponding to a tuple in $\mathcal{W}$. We say that partition $\mathcal{A}$ and instance I are *characterized* by $\bar{z} = (z_1, z_2, \ldots, z_t)$ if

$$\mathcal{A} = (x_k, x_1, x_1, \ldots, x_1) - V_1^{z_1} - V_2^{z_2} - \cdots - V_t^{z_t},$$

where subset V_j is defined by $\bar{b}_j$. In addition, if for instance I, partition $\mathcal{A}^{\mathrm{MON}}$ is given by (10) with $m_l = 0$ and $m \geq k-1$, i.e., $(\mathcal{A}, I) \in \Gamma^{(2)}$ for any $0 = x_1 \leq x_2 \leq \cdots \leq x_k$, then $\bar{z}$ is called *feasible*.

Now, it can be verified that the following formulation, in which we indicate two related subproblems, describes the problem of determining the performance ratio given by (13).

$$\left.\begin{array}{ll} \text{Maximize} & \dfrac{x_k + x_{k-1} + \cdots + x_1}{f_I(\mathcal{A})}, \\ \text{such that} & \mathcal{A}, I \text{ are characterized by } \bar{z} \\ & 0 = x_1 \leq x_2 \leq \cdots \leq x_k \\ & \bar{x} \geq \bar{0} \end{array}\right\} \text{Subproblem 2}$$
$$\left.\begin{array}{l} \bar{z} \text{ feasible} \\ \bar{z} \geq \bar{0},\ \bar{z} \text{ integer} \end{array}\right\} \text{Subproblem 1.}$$

The remainder of this section is organized as follows. First, we formulate Subproblem 1 as an Integer Linear Programming (ILP) problem and Subproblem 2 as a programming problem that is linear in $\bar{x}$ but conditional in $\bar{z}$. By replacing the conditional constraints by linear constraints at the cost of binary variables, we obtain an MILP formulation of the problem to determine the performance ratio for a given k. Finally, we discuss some results obtained by solving this MILP problem.

Subproblem 1. We will derive linear constraints for determining whether a given $\bar{z} \in \mathbb{N}^t$ is feasible, which yields an ILP formulation of Subproblem 1. Let problem instance I and partition $\mathcal{A}$ be characterized by $\bar{z}$. By definition, $\bar{z}$ is feasible if and only if the following two conditions are satisfied.

1. $m \geq k-1$, i.e., $\sum_{j=1}^{t} z_j \geq k-2$ as $m = 1 + \sum_{j=1}^{t} z_j$, and
2. $\mathcal{A}$ contains x_i once for $i = k$, $2m-1$ times for $i = k-1$, and m times for $1 \leq i \leq k-2$. This is equivalent to stating that $\mathcal{A}^{\mathrm{MON}}$ is given by $(x_k, x_{k-1}, \ldots, x_1) - (x_{k-1}, x_{k-1}, x_{k-2}, \ldots, x_1)^{m-1}$.

Consider Condition 2. By definition, we have that $\mathcal{A}$ contains x_k exactly once. Furthermore, to check the number of occurrences of the remaining $k-1$ variables, it suffices to check the number of occurrences of only $k-2$ of them. Hence, Condition 2 is equivalent to the condition that $\mathcal{A}$ contains $2m-1$ times x_{k-1} and m times x_i for $2 \le i \le k-2$. Since the total number of occurrences of x_i in $\mathcal{A}$ is given by $\sum_{j=1}^{t} z_j \cdot b_{ij}$, by definition, the condition is formalized by $\sum_{j=1}^{t} z_j \cdot b_{k-1,j} = 2m-1$ and $\sum_{j=1}^{t} z_j \cdot b_{ij} = m$ for $1 \le i \le k-2$. As a result, $\bar{z}$ is feasible, i.e., Conditions 1 and 2 hold, if and only if the linear constraints

$$\sum_{j=1}^{t} z_j \ge k-2$$
$$D\bar{z} = \bar{1},$$

are satisfied, where matrix D is given by

$$\begin{pmatrix} b_{21}-1 & b_{22}-1 & \cdots & b_{2t}-1 \\ b_{31}-1 & b_{32}-1 & \cdots & b_{3t}-1 \\ \vdots & & & \\ b_{k-2,1}-1 & b_{k-2,2}-1 & \cdots & b_{k-2,t}-1 \\ b_{k-1,1}-2 & b_{k-1,2}-2 & \cdots & b_{k-1,t}-2 \end{pmatrix}.$$

Subproblem 2. Next, we write Subproblem 2 as a programming problem with constraints that are conditional in $\bar{z}$ and linear in all other variables. We start with eliminating $f_I(\mathcal{A})$ from the problem formulation.

If partition $\mathcal{A}$ and instance I are characterized by $\bar{z}$, then $\mathcal{A}$ contains subset V_j if $z_j > 0$ and it does not contain V_j if $z_j = 0$. Hence, $f_I(\mathcal{A})$ equals the maximum sum of subset $x_k, x_1, x_1, \ldots, x_1$ and of any V_j with $z_j > 0$. As $x_1 = 0$, the sum of V_j is given by $\left(\sum_{i=2}^{k-1} b_{ij} x_i\right)$ and the sum of subset $x_k, x_1, x_1, \ldots, x_1$ by x_k. Hence, $f_I(\mathcal{A})$ is given by the minimum C satisfying

$$\sum_{i=2}^{k-1} b_{ij} x_i \le C, \quad \text{for all } j \text{ with } z_j > 0$$
$$x_k \le C.$$

As a result, Subproblem 2 is equivalent to

$$\begin{array}{lll} \text{Maximize} & \dfrac{x_k + x_{k-1} + \cdots + x_2}{C}, & \\ \text{such that} & \left(\displaystyle\sum_{i=2}^{k-1} b_{ij} x_i / C\right) \le 1, & \text{for all } 1 \le j \le t \text{ with } z_j > 0 \\ & x_k / C \le 1 & \\ & 0 \le x_2 \le x_3 \le \cdots \le x_k & \\ & x_i \ge 0, & \text{for all } 1 < i \le k \\ & C > 0. & \end{array}$$

We introduce new decision variables y_j with $1 < j < k$, which we substitute for x_j/C, i.e., y_j represents $x_j/f_I(\mathcal{A})$. Hence, instead of having a formulation that depends on both the exact cost of partition $\mathcal{A}$ and the exact values of $x_2, x_3, \ldots, x_k$, we have a formulation that only depends on the values of $x_2, x_3, \ldots, x_k$ expressed as a fraction of the cost of partition $\mathcal{A}$. We now get the following formalization of Subproblem 2. We thereby use that $y_k = x_k/C = 1$ in an optimal solution, which can easily be verified.

$$\begin{array}{lll} \text{Maximize} & 1 + \displaystyle\sum_{i=2}^{k-1} y_i, & \\ \text{such that} & \left(\displaystyle\sum_{i=2}^{k-1} b_{ij} y_i\right) \le 1, & \text{for all } j \text{ with } z_j > 0 \\ & 0 \le y_2 \le y_3 \le \cdots \le y_{k-1} \le 1 & \\ & y_i \ge 0, & \text{for all } 1 < i < k. \end{array}$$

To illustrate the first type of constraint, assume that $V_j = x_3\,x_3\,x_2\,x_2\,x_2$ occurs in partition $\mathcal{A}$ characterized by $\bar{z}$, i.e., $z_j > 0$. Then the constraint $\sum_{i=2}^{k-1} b_{ij} y_i \leq 1$ corresponds to $2y_3 + 3y_2 \leq 1$. Hence, the constraint gives a necessary condition on y_3 and y_2 to represent values x_3 and x_2 as a fraction of the cost of partition $\mathcal{A}$.

Subproblems 1 *and* 2 *combined.* The following problem formulation combines the derived constraints for Subproblems 1 and 2. In the formulation the integrality constraint on $\bar{z}$ is removed. This is allowed by Lemma 9 given below. For a proof of this lemma, we refer to Michiels [4].

$$
\begin{aligned}
\text{Maximize} \quad & 1 + \sum_{i=2}^{k-1} y_i, & & \\
\text{such that} \quad & \left(\sum_{i=2}^{k-1} b_{ij} y_i \right) \leq 1, & & \text{for all } 1 \leq j \leq t \text{ with } z_j > 0 \\
& \sum_{j=1}^{t} z_j \geq k-2 & & \\
& D\bar{z} = \bar{1} & & \\
& 0 \leq y_2 \leq y_3 \leq \cdots \leq y_{k-1} \leq 1 & & \\
& y_i \geq 0, & & \text{for all } 1 < i < k \\
& \bar{z} \geq \bar{0}. & &
\end{aligned}
\tag{P}
$$

Lemma 9. *Let real numbers $y_2, y_3, \ldots, y_{k-1}$ and real vector $\bar{z}$ define a solution of* (P)*. Then an integer $\bar{z}'$ exists, such that $y_2, y_3, \ldots, y_{k-1}$ and $\bar{z}'$ define a solution of* (P)*.*

We now show how (P) can be translated into an MILP problem, such that the problem can be solved by an MILP solver. However, one has to be careful when using a standard MILP solver since generally LP solvers, on which MILP solvers are based, are not exact due to rounding errors. Although in practice these rounding errors may have no effect on the outcome of the solver, in our case we need to use provably correct or so-called exact LP solvers since we use the outcome to establish performance ratios. Some research has been done on exact LP solvers [31,32]. However, these solvers are by far not as fast as standard solvers. In [4] we present an efficient dedicated algorithm for solving (P) that is reliable, although it makes use of an unreliable LP solver. Here, we restrict ourselves to showing how (P) can be solved by an exact MILP solver by formulating (P) as a MILP problem.

In order to do so, we have to eliminate the conditional constraints. To this end, we apply the approach presented in e.g. [33]. The approach needs an upper bound on z_j. Lemma 10 below gives such a bound. For $k = 4, 5, 6$, and 7 the upper bound equals 59, 529, $6.3 \cdot 10^3$, and $9.2 \cdot 10^4$, respectively. We prove the lemma in [4].

Lemma 10. *Adding the constraint*

$$z_j \leq (k^2 - k + 3)^{\frac{k-1}{2}}$$

to problem (P) *for some or all j with $1 \leq j \leq t$ does not affect its optimal cost.*

We are now able to eliminate the conditional constraints from (P). Consider the constraint

$$\sum_{i=2}^{k-1} b_{ij} y_i \leq 1 \quad \text{if } z_j > 0.$$

This constraint is equivalent to the either–or constraint

$$\sum_{i=2}^{k-1} b_{ij} y_i \leq 1 \quad \text{or} \quad z_j \leq 0. \tag{14}$$

We introduce the binary variable β_j, and we define Z as an upper bound on the second constraint, i.e., Z is an upper bound on z_j. By Lemma 10, $(k^2 - k + 3)^{(k-1)/2}$ is a valid value for Z. Furthermore, by the definition of b_{ij} and since $y_j \leq 1$, we have that k is a valid upper bound on the first constraint in (14). Now, (14) is equivalent to

$$\sum_{i=2}^{k-1} b_{ij} y_i \leq 1 + (k-1)\beta_j \quad \text{and} \quad z_j \leq (1 - \beta_j) Z.$$

Table 1
Size of problem (P′).

k	Constraints	Real variables	Binary variables
4	36	17	15
5	120	59	56
6	430	214	210
7	1596	797	792

Hence, we get the following MILP problem for finding the performance ratio of BLDM for fixed $k \geq 4$.

$$
\begin{array}{lll}
\text{Maximize} & 1 + \sum_{i=2}^{k-1} y_i, & \\
\text{such that} & \left(\sum_{i=2}^{k-1} b_{ij} y_i \right) \leq 1 + (k-1)\beta_j, & \text{for all } 1 \leq j \leq t \\
 & z_j \leq (1-\beta_j) Z, & \text{for all } 1 \leq j \leq t \\
 & \sum_{j=1}^{t} z_j \geq k-2 & \\
 & D\bar{z} = \bar{1} & \\
 & 0 \leq y_2 \leq y_3 \leq \cdots \leq y_{k-1} \leq 1 & \\
 & y_i \geq 0, & \text{for all } 1 < i < k \\
 & \bar{z} \geq \bar{0} & \\
 & \bar{\beta} \in \{0,1\}^t. &
\end{array}
\qquad (\mathrm{P}')
$$

Regarding the size of (P′) it can be verified that the problem consists of $t + k - 2$ real variables, t binary variables, and $2t + 2k - 2$ constraints, not counting the constraints $y_i \geq 0$ with $1 < i < k$ and $\bar{z} \geq \bar{0}$. Furthermore, t is by definition the number of $(k-1)$-tuples $\bar{b}_j$ with $\sum_{i=1}^{k-1} b_{ij} = k$. Hence, we have

$$
t = \binom{2k-2}{k}.
$$

Table 1 indicates the size of (P′) for $k = 4, 5, 6$, and 7.

The above MILP problem can be solved by an exact MILP solver. By using the more efficient dedicated algorithm presented in [4] we obtained for $k = 4, 5, 6$, and 7 the performance ratios 19/12, 103/60, 643/360, and 4603/2520, respectively. These performance ratios are equal to the lower bound presented in Section 4, i.e., this lower bound is tight for $k \leq 7$.

The results derived are summarized in the theorem below. For the sake of completeness, we also add the performance ratio of 4/3 for $k = 3$. The proof of this ratio, which is given in [4], is omitted in this paper because it is complicated by case distinctions and does not introduce new ideas.

Theorem 11. *For given $k \geq 3$, the performance ratio of* BLDM *is bounded from below by $2 - \sum_{i=0}^{k-1} \frac{i!}{k!}$. For $k \leq 7$ this bound is tight, i.e., it gives the performance ratio of* BLDM. *For $k > 7$ an upper bound is given by $2 - \frac{1}{k-1}$.*

7. Conclusion

In this paper, we analyzed the performance ratio of BLDM as a function of the cardinality k of the subsets. We proved that a lower bound on the performance ratio is given by $2 - \sum_{i=0}^{k-1} \frac{i!}{k!}$. Furthermore, we formulated the problem of deriving the performance ratios as an MILP problem. For $k \leq 7$ this MILP problem can be solved by an exact MILP solver. This results in a computer-assisted proof of the tightness of the lower bound for $k \leq 7$. For $k > 7$ we proved that an upper bound on the ratio is given by $2 - \frac{1}{k-1}$.

We show in [34] that in addition to number partitioning and balanced number partitioning, the Differencing Method can also be applied to the Min–Max Subsequence problem. In this problem, one has to order a collection of numbers such that the maximum sum of k successive numbers is minimized. In [4] we prove that by a similar analysis as the one presented in this paper, we can derive worst-case performance results for the Differencing Method when applied to the Min–Max Subsequence problem.

For the three problems mentioned, the Differencing Method outperforms all known practical polynomial-time algorithms from an average-case perspective. Simulations also show promising results for the case that the Differencing Method is applied to the classical bin packing problem. A further study on this is left as an interesting research direction.

Appendix. Proof of Lemma 2

Lemma 2. *Solution $\mathcal{A}$ defined by (5) is a feasible solution.*

Proof. In order to show that $\mathcal{A}$ is a feasible solution, we need to show that

(i) $\sum_{i=1}^{k-2} z_i^{(k)} = m^{(k)} - 1$,
(ii) $z_i^{(k)}$ is integral, and
(iii) each variable x_i has the correct number of occurrences.

(i) We use induction on k. The validity for $k = 4$ is easily verified, since $z_1^{(4)} + z_2^{(4)} = 5 + 2 = m^{(4)} - 1$. Assuming, using the induction hypothesis that $\sum_{i=1}^{\ell-2} z_i^{(\ell)} = m^{(\ell)} - 1$, we will now derive that $\sum_{i=1}^{\ell-1} z_i^{(\ell+1)} = m^{(\ell+1)} - 1$. First observe that, using (2),

$$m^{(\ell+1)} = 1 + \frac{1}{2}\sum_{i=2}^{\ell}\sum_{j=1}^{i-1}(\ell+1-j)! = \sum_{i=2}^{\ell}\sum_{j=0}^{i-2}(\ell-j)! = \sum_{i=1}^{\ell-1}\sum_{j=0}^{i-1}(\ell-j)!.$$

Splitting off $i = 2$ and $j = 1$ gives that the right-hand side expression equals $m^{(\ell)} + \frac{1}{2}(l-1)l!$. Hence,

$$m^{(\ell+1)} - m^{(\ell)} = \frac{1}{2}(\ell-1)\ell!. \tag{15}$$

We also have, using (4),

$$z_1^{(\ell+1)} - z_1^{(\ell)} = \frac{2m^{(\ell+1)}-1}{3} - \frac{2m^{(\ell)}-1}{3} = \frac{2}{3}(m^{(\ell+1)} - m^{(\ell)}), \tag{16}$$

$$z_{\ell-1}^{(\ell+1)} = \frac{(\ell+1)! - (\ell-1)! - \ell!}{(\ell+1)(\ell-1)!} = \ell - 1, \tag{17}$$

and, for each p with $2 \le p \le k-2$

$$z_p^{(\ell+1)} - z_p^{(\ell)} = \frac{(\ell+1)! - \sum_{j=p}^{\ell} j!}{(p+2)p!} - \frac{\ell! - \sum_{j=p}^{\ell-1} j!}{(p+2)p!} = \frac{(\ell+1)! - 2\ell!}{(p+2)p!} = \frac{(\ell-1)\ell!}{(p+2)p!}. \tag{18}$$

Using the induction hypothesis, the equality $\sum_{i=1}^{\ell-1} z_i^{(\ell+1)} = m^{(\ell+1)} - 1$, which we need to prove, is equivalent to

$$\sum_{i=1}^{\ell-1} z_i^{(\ell+1)} - \sum_{i=1}^{\ell-2} z_i^{(\ell)} = m^{(\ell+1)} - m^{(\ell)} \Leftrightarrow z_{\ell-1}^{(\ell+1)} + \sum_{i=1}^{\ell-2}(z_i^{(\ell+1)} - z_i^{(\ell)}) = m^{(\ell+1)} - m^{(\ell)}.$$

Using (16), this latter equality can be rewritten to

$$z_{\ell-1}^{(\ell+1)} + \sum_{i=2}^{\ell-2}(z_i^{(\ell+1)} - z_i^{(\ell)}) = \frac{1}{3}(m^{(\ell+1)} - m^{(\ell)}),$$

which by using (15), (17) and (18) can be rewritten to

$$\ell - 1 + \sum_{i=2}^{\ell-2}\frac{(\ell-1)\ell!}{(i+2)i!} = \frac{1}{6}(\ell-1)\ell! \Leftrightarrow (\ell-1)\ell!\sum_{i=2}^{\ell-2}\frac{1}{(i+2)i!} = \frac{1}{6}(\ell-1)\ell! - \ell + 1$$
$$\Leftrightarrow \sum_{i=2}^{\ell-2}\frac{1}{(i+2)i!} = \frac{1}{6} - \frac{1}{\ell!}.$$

The latter equality can easily be proved by induction on l. This proves that $\sum_{i=1}^{\ell-1} z_i^{(\ell+1)} = m^{(\ell+1)} - 1$.

(ii) Let $z_i^{(k)}$ with $1 \le i \le k-2$ be defined as in (4). Then each $z_i^{(k)}$ is integral as can be seen as follows. First, consider $z_1^{(k)} = \frac{2m^{(k)}-1}{3}$. Rewriting the right-hand side expression in (2) gives

$$m^{(k)} = 1 + \frac{1}{2}\sum_{j=2}^{k-1}(j-1)j! = 2 + \frac{1}{2}\sum_{j=3}^{k-1}(j-1)j!.$$

Hence, $2m^{(k)} - 1$ is given by $3 + \sum_{j=3}^{k-1}(j-1)j!$. This expression is divisible by 3 since $j!$ is divisible by 3 for all $j \ge 3$. This proves that $z_1^{(k)}$ is integral.

Next, consider a value $z_i^{(k)}$ with $i > 1$. Each term $j!$ with $j \geq i+2$ is divisible by $(i+2)i!$. Furthermore, we have $i! + (i+1)! = (i+2)i!$ and $k \geq i+2$. These observations imply that $k! - \sum_{j=i}^{k-1} j!$ is divisible by $(i+2)i!$. By (4) this proves that $z_i^{(k)}$ is integral.

(iii) It remains to show that x_k occurs once, x_{k-1} occurs $2m^{(k)} - 1$ times, and x_i with $1 \leq i \leq k-2$ occurs $m^{(k)}$ times in $\mathcal{A}$. For x_k this is obviously true. The variable x_{k-1} occurs 3 times in V_1 and it does not occur in any other set V_j. As $z_1^{(k)} = \frac{2m-1}{3}$ this implies that also x_{k-1} has the correct number of occurrences. Next, consider x_i with $2 \leq i \leq k-2$. We have that x_i occurs $k-i+2$ times in V_{k-i} and once in V_j with $j \leq k-i-2$. It does not occur in any other subset V_j. Hence, the number of occurrences of x_i in $\mathcal{A}$ is given by

$$(k-i+2)z_{k-i}^{(k)} + \sum_{j=1}^{k-i-2} z_j^{(k)}. \tag{19}$$

In order to show that the latter expression equals $m^{(k)}$, we will first prove the following equality for each $\hat{\imath}$ with $\hat{\imath} \leq k-2$

$$\sum_{j=1}^{\hat{\imath}} z_j^{(k)} = m^{(k)} + \frac{\sum_{j=\hat{\imath}+2}^{k-1} j! - k!}{(\hat{\imath}+2)!}. \tag{20}$$

We use downward induction to establish (20). As we proved (i), the equality holds for $\hat{\imath} = k-2$. Now, assuming (20) holds for $\hat{\imath} = \ell$, we prove that it holds for $\hat{\imath} = \ell - 1$ as follows.

$$\begin{aligned}
\sum_{j=1}^{\ell-1} z_j^{(k)} &= \sum_{j=1}^{\ell} z_j^{(k)} - z_\ell^{(k)} \\
&= m^{(k)} + \frac{\sum_{j=\ell+2}^{k-1} j! - k!}{(\ell+2)!} - \frac{k! - \sum_{j=\ell}^{k-1} j!}{(\ell+2)\ell!} \\
&= m^{(k)} + \frac{\sum_{j=\ell+2}^{k-1} j! - (\ell+2)k! + (\ell+1)\sum_{j=\ell}^{k-1} j!}{(\ell+2)!} \\
&= m^{(k)} + \frac{(\ell+2)\sum_{j=\ell+2}^{k-1} j! + (\ell+1)(\ell! + (l+1)!) - (\ell+2)k!}{(\ell+2)!} \\
&= m^{(k)} + \frac{(\ell+2)\sum_{j=\ell+2}^{k-1} j! + (\ell+2)(\ell+1)! - (\ell+2)k!}{(\ell+2)!} \\
&= m^{(k)} + \frac{\sum_{j=\ell+1}^{k-1} j! - k!}{(\ell+1)!}.
\end{aligned}$$

This proves the correctness of (20). We now return to the number of occurrences of x_i in $\mathcal{A}$. As indicated, this number is given by (19). Using (20), we can rewrite this expression to

$$(k-i+2)z_{k-i}^{(k)} + m^{(k)} + \frac{\sum_{j=k-i}^{k-1} j! - k!}{(k-i)!} = (k-i+2)z_{k-i}^{(k)} + m^{(k)} - (k-i+2)z_{k-i}^{(k)} = m^{(k)}.$$

This means that $\mathcal{A}$ has the correct number of occurrences of any x_i with $2 \leq i \leq k$. Furthermore, as the total number of elements in $\mathcal{A}$ and I_k is the same, i.e., km, we automatically get that the number of occurrences of x_1 must also be correct. This proves the feasibility of $\mathcal{A}$. □

References

[1] N. Karmarkar, R.M. Karp, The Differencing Method of set partitioning. Technical Report UCB/CSD 82/113, University of California, Berkeley, 1982.
[2] R.L. Graham, Bounds on multiprocessing timing anomalies, SIAM Journal on Applied Mathematics 17 (2) (1969) 416–429.
[3] E.G. Coffman Jr., M.R. Garey, D.S. Johnson, An application of bin-packing to multiprocessor scheduling, SIAM Journal on Computing 7 (1) (1978) 1–17.

[4] W. Michiels, Performance Ratios for Differencing Methods. Ph.D. Thesis, Technische Universiteit Eindhoven, 2004.
[5] S. Boettcher, S. Mertens, Analysis of the Karmarkar-Karp differencing algorithm, The European Physical Journal B 65 (2008) 131–140.
[6] E.G. Coffman Jr., W. Whitt Jr., Recent asymptotic results in the probabilistic analysis of schedule makespans, in: P. Chretienne, E.G. Coffman, J.K. Lenstra, Z. Liu (Eds.), Scheduling Theory and its Applications, Wiley, 1995, pp. 15–31.
[7] L.H. Tasi, The modified differencing method for the set partitioning problem with cardinality constraints, Discrete Applied Mathematics 63 (1995) 175–180.
[8] B. Yakir, The differencing algorithm LDM for partitioning: a proof of a conjecture of Karmarkar and Karp, Mathematics of Operations Research 21 (1) (1996) 85–99.
[9] L. Babel, H. Kellerer, V. Kotov, The k-partitioning problem, Mathematical Methods of Operations Research 47 (1998) 59–82.
[10] L.H. Tsai, Asymptotic analysis of an algorithm for balanced parallel processor scheduling, SIAM Journal on Computing 21 (1) (1992) 59–64.
[11] M.O. Ball, M.J. Magazine, Sequencing of insertions in printed circuit board assembly, Operations Research 36 (2) (1988) 192–201.
[12] L.H. Tsai, The loading and scheduling problems in flexible manufacuring systems. Ph.D. Thesis, University of California, Berkeley, 1987.
[13] C. McDiarmid, Pattern minimisation in cutting stock problems, Discrete Applied Mathematics 98 (1999) 121–130.
[14] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[15] D.S. Hochbaum, D.B. Shmoys, Using dual approximation algorithms for scheduling problems: theoretical and practical results, Journal of the ACM 34 (1987) 144–162.
[16] K. Appel, W. Haken, Every planar map is four colorable. part I: discharging, Illinois Journal of Mathematics 21 (1977) 429–490.
[17] K. Appel, W. Haken, Every Planar Map is Four Colorable, in: American Mathematical Society, Contemporary Mathematics, vol. 98, 1989.
[18] K. Appel, W. Haken, J. Koch, Every planar map is four colorable. part II: reducibility, Illinois Journal of Mathematics 21 (1977) 491–567.
[19] U. Zwick, Computer assisted proof of optimal approximability results. in: Proceedings of the 13th Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 496–505, San Francisco, California, 2002.
[20] E.G. Coffman Jr., D.S. Johnson, P.W. Shor, R.R. Weber, Markov chains, computer proofs, and average-case analysis of best fit bin packing. in: Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, pp. 412–421, 1993.
[21] R. Bubley, M. Dyer, C. Greenhill, Beating the 2δ bound for approximately counting colourings: a computer-assisted proof of rapid mixing. in: Proceedings of the 9th Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 355–363, San Francisco, California, 1998.
[22] T.C. Hales, A proof of the kepler conjecture, Annals of Mathematics 162 (3) (2005) 1065–1185.
[23] H. Kellerer, G.J. Woeginger, A tight bound for 3-partitioning, Discrete Applied Mathematics 45 (1993) 249–259.
[24] H. Kellerer, V. Kotov, A 7/6-approximation algorithm for 3-partitioning and its application to multiprocessor scheduling, INFOR 37 (1) (1999) 48–56.
[25] M. Fischetti, S. Martello, Worst-case analysis of the differencing method for the partition problem, Mathematical Programming 37 (1987) 117–120.
[26] W. Michiels, J. Korst, E. Aarts, J. van Leeuwen, Performance ratios of the Karmarkar–Karp differencing method, Journal of Combinatorial Optimization 13 (2007) 19–32.
[27] E.G. Coffman Jr., G.N. Frederickson, G.S. Lueker, A note on expected makespans for largest-first sequences of independent tasks on two processors, Mathematics of Operations Research 9 (1984) 260–266.
[28] G.S. Lueker, A note on the average-case behavior of a simple differencing method for partitioning, Operations Research Letters 6 (6) (1987) 285–287.
[29] R.E. Korf, A complete anytime algorithm for number partitioning, Artificial Intelligence 106 (1998) 181–203.
[30] S. Mertens, A complete anytime algorithm for balanced number partitioning, 1999. (preprint).
[31] M. Dhiflaoui, S. Funke, C. Kwappik, K. Mehlhorn, M. Seel, E. Schömer, R. Schulte, D. Weber, Certifying and repairing solutions to large LPs: how good are LP-solvers. in: Proceedings of the 14th Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 255–256, Baltimore, MD, 2003.
[32] B. Gärtner, Exact arithmetic at low cost: a case study in linear programming. in: Proceedings of the 9th ACM–SIAM Symposium on Discrete Algorithms, pp. 157–166, San Francisco, California, 1998.
[33] H.P. Williams, Model Building in Mathematical Programming, Wiley, 1978.
[34] W. Michiels, J. Korst, Min–max subsequence problems in multi-zone disk recording, Journal of Scheduling 4 (5) (2001) 271–283.
[35] W. Michiels, J. Korst, E. Aarts, J. van Leeuwen, Performance ratios for the Differencing Method applied to the balanced number partitioning problem. in: Proceedings of the 20th International Symposium on Theoretical Aspects of Computer Science (STACS'03), pp. 583–595, Berlin, 2003.